

SERVER, DEVICE, CLIENT, INFORMATION PROCESSING
METHOD OF SERVER, INFORMATION PROCESSING METHOD
OF DEVICE, INFORMATION PROCESSING METHOD OF
CLIENT, INFORMATION PROCESSING PROGRAM, AND
5 MEMORY MEDIUM

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates to a network system in which
10 a server, a device, and a client communicate with each
other through a predetermined communication medium and
execute data processes. More particularly, the
invention relates to the server, device, and client in
the network system, a data processing method of the
15 server, a data processing method of the device, and a
data processing method of the client.

Related Background Art

Hitherto, a method called a directory service has
been provided as a method whereby a computer of the
20 user efficiently finds out and uses various resources
(printer, server, scanner, and the like) connected to a
network.

The directory service is, so to speak, a phone
book regarding the network and used for storing various
25 information. As a specific example of a directory
system using the directory service, for instance, there
is an LDAP (Light Weight Directory Access Protocol).

The regulations of the LDAP have been disclosed in RFC (Request For Commits) 1777 as a standard issued by IETF (Internet Engineering Task Force).

5 As a manual of the LDAP, for example, "LDAP Internet Directory Application Programming" has been issued by Plentice Hall Co., Ltd. on Nov. 1, 1997.

10 A computer of the user retrieves the devices connected to, for example, the network by using the directory service, so that it can obtain a list of network addresses of the devices which can be used on the network.

SUMMARY OF THE INVENTION

15 In the above conventional method, however, when considering a network printer as an example, if various function conditions such as 2-page printing mode (mode to print two pages onto one sheet of paper), OHP inserting mode, duplex mode, stapling, paper size, and the like are designated and a retrieval is performed, 20 the function conditions and ability information of the devices are compared, and only the devices in which they perfectly coincide are retrieved or inadaptivity items of the devices in which they do not perfectly coincide are notified.

25 The invention is made to solve the above problem and it is an object of the invention to provide a server, a client, a device, a data processing method of

the server, a data processing method of the client, a data processing method of the device, and a memory medium, in which devices connected to a network and ability information of device drivers are combined and registered as abilities of the devices into the server, thereby presenting adaptivities of the selectable devices according to a device retrieval condition from the client to the user, widening a selection range of the devices which can be selected by the user in the whole system, and enabling the user to freely select a desired device from a plurality of devices which conforms with the retrieval condition.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram for explaining a construction of a network system to which a server, a client, and a device can be applied and which shows an embodiment of the invention;

Fig. 2 is a diagram showing a specific construction of a network on which a device retrieval system to which the network system shown in Fig. 1 can be applied can operate;

Fig. 3 is a schematic constructional block diagram showing an internal construction of a general personal computer;

Fig. 4 is a diagram for explaining address information and attribute information of each of the

devices which are managed by the server shown in Fig. 2;

Fig. 5 is a diagram showing values which can be used as items serving as a retrieval condition in case of retrieving from the client shown in Fig. 2;

Fig. 6 is a diagram showing a retrieval condition input screen in case of retrieving from the client shown in Fig. 2;

Fig. 7 is a diagram showing a retrieval condition input screen in case of retrieving from the client shown in Fig. 2;

Fig. 8 is a diagram showing an example of a format of data which is sent and received among the server, client, and devices shown in Fig. 2;

Fig. 9 is a diagram showing an example of ability information of the device which is managed by the server shown in Fig. 2;

Fig. 10 is a diagram showing ability information of a device driver which is managed by the server shown in Fig. 2;

Fig. 11 is a diagram showing ability information in which the ability information possessed by the device shown in Fig. 1 and the ability information of the device driver are combined;

Fig. 12 is a diagram showing an example of a format showing adaptivity results which are sent from the server to the client shown in Fig. 2;

Fig. 13 is a diagram showing an example of a retrieval result informing screen which is displayed to the client shown in Fig. 2;

Fig. 14 is a block diagram showing an internal construction of a color printer shown in Fig. 2;

Fig. 15 is a flowchart showing an example of a first data processing procedure in the device according to the invention;

Fig. 16 is a flowchart showing an example of a first data processing procedure in the server according to the invention;

Fig. 17 is a flowchart showing an example of a second data processing procedure in the server according to the invention;

Fig. 18 is a flowchart showing an example of a third data processing procedure in the server according to the invention;

Fig. 19 is a flowchart showing an example of a first data processing procedure in the client according to the invention;

Fig. 20 is a flowchart showing an example of a second data processing procedure in the device according to the invention;

Fig. 21 is a flowchart showing an example of a second data processing procedure in the client according to the invention;

Fig. 22 is a diagram for explaining a memory map

of a memory medium for storing various control programs according to the invention;

Fig. 23 is a flowchart showing an example of a device driver retrieval processing procedure in the server and a client terminal apparatus according to the invention; and

Fig. 24 is a diagram for explaining a memory map of a memory medium for storing various data processing programs which can be read out by a network system including the server, client, device, and the like according to the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

An explanation will now be made hereinbelow with respect to an embodiment of a server, a client, and a device on a network, a device retrieval system (network system comprising the server, client, and device), a data processing method of the server, a data processing method of the client, a data processing method of the device, and a memory medium according to the invention with reference to the drawings.

Fig. 1 is a block diagram for explaining a construction of the network system which is realized by the server, client, and device and shows an embodiment of the invention.

In Fig. 1, a server 10 which functions as a retrieval server, a client 20, and a device 30 are

connected to a network 40 such as Ethernet, USB, or the like, thereby constructing the network system. The device 30 is, for example, an information processing apparatus such as printer, facsimile, scanner, or the like and, usually, a plurality of devices 30 are connected.

In the embodiment, a workstation is used as a server 10 and has: a manager 11; a memory 11a for performing an arithmetic operation and storing information; a retrieval unit 12; a receiver 13; a transmitter 14; an ability information table 15; and a device driver storage 16.

The manager 11 manages devices connected to the network 40 by the ability information table 15 in which connection information and ability information of the device 30 have been stored, manages a device driver of the connected device, and downloads the device driver in response to a request from the client. Further, when an inquiry regarding the ability information of the device driver is received from the device 30, the manager 11 returns the ability information of the device driver.

The device driver is a program which is executed by the server and client and is used for controlling the device. For example, a printer driver as an example of the device driver receives document data from a document generating program or a graphics

forming program which is executed by the client and converts the document data into print data which can be processed by the device. If a printer does not have an N-up printing function, the printer driver generates
5 print data so that N pages of the document data are allocated to one page of the print data.

The receiver 13 receives a retrieval request from the client 20. The retrieval unit 12 compares the retrieval request received by the receiver 13 with the
10 ability information of each device connected to the network 40 through the ability information table 15, thereby calculating an adaptivity between each device and the retrieval condition. The adaptivity is sent through the transmitter 14 to the client 20 or the like
15 to which the retrieval request has been issued.

The ability information of each device connected to the network 40 has been stored in the ability information table 15 at the time of connection of the device or the turn-on of a power source. When the
20 ability information is changed due to the updating of the device driver, it is also restored.

The device driver storage 16 stores the driver of the device connected to the network 40 as necessary. By previously storing the device driver into the server
25 10, a service for presenting the driver in the case where the client 20 does not have the driver of the adapted device 30 can be performed.

A personal computer is used as a client 20 and has: a retrieval request unit 21; a manager 22; a device driver storage 23; a display controller 24; and a display 25.

5 In the client 20, the retrieval request unit 21 requests the server 10 to retrieve all of the devices which can be used on the network 40 including the device 30.

10 The manager 22 holds a result of the retrieval request. When results regarding all of the devices are obtained, the manager 22 notifies the display controller 24 of them. When an inquiry about the ability information of the device driver is received from the device 30, the manager 22 returns the ability
15 information of the driver.

 The device driver storage 23 is means for storing the driver for using the device connected to the network 40. The display controller 24 sequentially displays adaptivity results onto a screen of the
20 display 25 in the order from the high adaptivity on the basis of the adaptivity results notified from the manager 22.

 The device 30 has a device driver storage 31, a manager 32, and an ability information table generator
25 33. When the device 30 is connected to the network 40, the ability information table is sent to the server 10.

 In the device 30, the manager 32 updates the

device driver and sends the ability information table to the server 10. It will be obviously understood that they can be integratedly constructed by a CPU, an MPU, a memory, and the like.

5 The ability information table generator 33 obtains the sum of the ability information of the device 30 and the ability information of the device driver at the time of power-on, thereby generating an ability information table.

10 When no device driver exists in the device 30, whether the driver of the device 30 exists in the server 10 or client 20 or not is confirmed. If it exists, the ability information is obtained and an ability information table is generated on the basis of
15 the obtained ability information and the ability information of the device 30.

Each function of a device retrieval system of the embodiment constructed as mentioned above will now be described in detail hereinbelow.

20 Fig. 2 is a diagram showing a specific construction of a network on which the device retrieval system to which the network system shown in Fig. 1 has been applied can operate.

25 In Fig. 2, reference numeral 213 denotes a color printer (color LBP); 214 an MFP (Multi Function Peripheral; although it is a copy machine, it can be used also as a network color printer, a scanner, or a

facsimile); and 215 a monochromatic printer.

Reference numeral 211 denotes a desktop PC (client). It is a PC (personal computer) which can execute a program of a network client. The desktop PC 211 is connected to a network 200 as a client mentioned above. As will be explained hereinlater, the desktop PC 211 has not only a function for issuing inquiry information regarding the device which satisfies desired conditions to the server connected on the network 200 but also a function for displaying retrieval results.

Reference numeral 210 denotes a workstation (WS) which can execute a program of the network server in the embodiment. The workstation (WS) 210 is connected as a server mentioned above to the network 200. As will be explained hereinlater, various information regarding the network devices 213 to 215 has been stored in the WS 210. The WS 210 has a function for receiving a device retrieval inquiry from the client 211 connected to the network 200 and returning its result.

Fig. 3 is a schematic constructional diagram showing an internal construction of a general personal computer. A fundamental internal construction of each of the desktop PC 211, server 210, and the like in Fig. 2 is as shown in the diagram.

In Fig. 3, reference numeral 300 denotes a PC or

WS on which client software or network server software
(hereinafter, they are collectively called network
device retrieval software) operates. The PC/WS 300 is
substantially equivalent to the desktop PC 211 or
5 server 210 in Fig. 2.

The PC 300 has a CPU 302 for executing the network
device retrieval software which has been stored in an
ROM 303 or a hard disk (HD) 311 or is supplied from a
floppy disk controller (FDC) 312. The PC 300
10 integratedly controls the devices connected to a system
bus 301.

Reference numeral 304 denotes an RAM which
functions as a main memory, a work area, or the like of
the CPU 302; 305 a keyboard controller (KBC) for
15 controlling an input of an instruction from a keyboard
(KB) 309; and 306 a CRT controller (CRTC) for
controlling a display of a CRT display (CRT) 310.

Reference numeral 307 denotes a disk controller
(DKC) for controlling accesses to the hard disk (HD)
20 311 for storing a boot program, various applications,
an edit file, a user file, a network management
program, and the like and to the floppy disk controller
(FDC) 312.

Reference numeral 308 denotes a network interface
25 card (NIC) for bidirectionally transmitting and
receiving data to/from a network printer, another
network apparatus, or another PC through an LAN 320.

Reference numeral 313 denotes a mouse controller (MC) for controlling a mouse (MS) 314. In the embodiment, the LAN 320 corresponds to the LAN 200 in Fig. 2.

5 Fig. 14 is a block diagram showing an internal construction of the color LBP 213 shown in Fig. 2.

As shown in Fig. 14, in the color LBP 213, a CPU 1402 for processing a program, an ROM 1403 in which the program has been stored, and an RAM 1404 serving as a
10 work area or a buffer area of the program are connected to a system bus 1401.

Reference numeral 1405 denotes an LBP engine controller (EC). An engine (LBP Engine) 1409 is connected through the controller 1405. Reference
15 numeral 1406 denotes a panel controller (PLC) for controlling input/output to/from a panel (PAENL) 1410 and managing the panel 1410.

The color LBP 213 in the embodiment has a hard disk (HD) 1411 and can temporarily spool print data
20 into the hard disk (HD) 1411, thereby making it possible to cope with a jobcopy.

Reference numeral 1407 denotes a disk controller for managing the HD 1411, and 1408 indicates a network interface controller (NIC). The color LBP is connected
25 to the network 320 through the NIC 1408.

Reference numeral 1412 denotes a non-volatile RAM (NVRAM) which can hold data even if a power source of

the printer 213 is turned off. In the embodiment, although layer type position information, attribute information, and the like are held in the NVRAM 1412, those information can be also held in the HD 1411 instead of the NVRAM 1412.

Fig. 4 is a diagram for explaining address information and attribute information of each device which is managed by the server 210 shown in Fig. 2. In the embodiment, the address information, attribute information, and the like of each device are totally called device information and, further, it is assumed that data which can be managed is included as necessary.

As shown in Fig. 4, in the embodiment, three devices are managed in total. In each device, information such as device name (NM) 401, IP address (ADD) 402 as a network address of the device, and kind of device (DV) 403 is managed. For example, in case of the device whose name is "LBP1110", the IP address is "111.22.33.44" and the kind of device is "printer".

Fig. 5 is a diagram showing values which can be used every item serving as a retrieval condition in case of retrieving from the client 211 shown in Fig. 2. For example, in an item 502, whether a duplex print is designated (ON) or not (OFF) can be selected. In an item 503, one of a 2-up print (two pages are printed onto one sheet of paper), a 4-up print (four pages are

printed onto one sheet of paper), and a 6-up print (six pages are printed onto one sheet of paper) can be designated. In an item 504, how many jobs are copied can be designated. In an item 505, how many pages are copied can be designated. In an item 506, whether an OHP insertion print is designated (ON) or not (OFF) can be selected. In an item 507, a resolution can be designated. In an item 508, the number of pages can be designated. In an item 509, one of the paper sizes A5, A4, and A3 can be designated.

In the client 211, the user can input a retrieval condition in a set range with respect to the items 502 to 509.

Figs. 6 and 7 are diagrams each showing a retrieval condition input screen in case of retrieving from the client 211 shown in Fig. 2.

In Fig. 6, entries of retrieval are inputted into items 601a, 601b, 601c, and 601d, and conditions corresponding to the entries are inputted into items 602a, 602b, 602c, and 602d.

The entries are different depending on the devices. The user selects the attribute information to be retrieved from the entries. The embodiment relates to a printer as shown in Fig. 5. For example, the items 502 to 509 of a retrieval condition 501 are inputted into the items 601a to 601d and their values are inputted into the items 602a to 602d. After the

retrieval condition was inputted, the user presses a retrieval start button 603. Although four items can be inputted per screen in the embodiment, all of the items in the retrieval condition 501 can be inputted by a
5 next page button 604.

Fig. 7 shows an example of a retrieval input. In this example, the device in which, as attribute information, the device can be set to "printer", page print designation (N-up) can be set to "4", the jobcopy
10 No. (JOB-COPY) can be set to "10", and the pagecopy No. (page-copy) can be set to "2" is retrieved.

Although not shown in Fig. 7, each item of the retrieval condition can be also set as follows. For example, duplex indicates both-sides, the paper size is
15 set to A4, the resolution is set to 600 dpi, the number of pages is set to 50, and the like. Thus, the device in which the duplex print can be performed, the print can be performed to the paper of the A4 size, and the resolution is equal to 600 dpi is retrieved.

Fig. 8 is a diagram showing an example of a format of data which is sent and received among the server
20 210, client 211, and devices 213 to 215 shown in Fig. 2.

In the diagram, reference numeral 801 denotes an
25 example of a format for allowing the client 211 to inquire of the server 210 about the retrieval condition shown in Fig. 7. When the data is transferred from the

client 211 to the server 210, the retrieval is performed. The format 801 in Fig. 8 corresponds to a case where the device DV is "printer", the duplex print is "ON (duplex is designated)", the N-up print is "4-
5 page print", JOBcopy is "10", Pagecopy is "2", the paper size is "A4", and the OHP insertion mode resolution is "600 dpi".

Reference numeral 802 denotes an example of a format for allowing the client 211 to obtain the device
10 information from the server 210. The data is transferred from the client 211 to the server 210. For example, Dev = status indicates that the statuses of the devices 213 to 215 are requested. In response to this request, the server 210 returns the status such as
15 ready, off-line, printing, or the like of each device. Dev = error indicates a conformation request to discriminate whether the device is in an error status or not. In response to this request, the server 210 returns factors of errors.

20 Reference numeral 803 shows an example of a format for allowing the devices 213 to 215 to identify the presence or absence of the device driver and the ability information to the server 210 and client 211. The data is transferred from the device to the server
25 210 or from the device to the client 211. Driver = LBP1110 is used for asking the presence or absence of the driver of the LBP 1110. Info = driverLBP1110 is

used for requesting for the ability information of the device driver of the LBP 1110.

Fig. 9 is a diagram showing an example of ability information of a certain device which is managed by the server 210 shown in Fig. 2.

In the diagram, ability information has been set for each of items 1101 to 1107. Each item shows as follows. That is, as an item 1101, duplex is "available". As an item 1102, N-up print is "unavailable". As an item 1103, JOBCopy is available to "10". As an item 1104, Pagecopy is available to "999". As an item 1105, paper size is "only A4". As an item 1106, OHP insertion is "unavailable". As an item 1107, resolution is "both 600/300 dpi are available".

Fig. 10 is a diagram showing ability information of a certain device driver which is managed by the server 210 shown in Fig. 2. The same component elements as those in Fig. 9 are designated by the same reference numerals.

In the diagram, ability information has been set for each of the items 1101 to 1107. For example, as an item 1101, duplex is "unavailable". As an item 1102, N-up print is "available". As an item 1103, JOBCopy is "available". As an item 1104, Pagecopy is "available". As an item 1105, paper size is "only A4". As an item 1106, OHP insertion is "available". As an item 1107,

resolution is "both 600/300 dpi are available".

Fig. 11 is a diagram showing ability information in which the ability information possessed by a certain device and the ability information of the device driver of such a device are combined. The ability information in Fig. 11 is based on the ability information in Fig. 9 and the ability information in Fig. 10. The ability information in Fig. 11 is managed in the ability information table 15.

In the diagram, the ability information has been set for each of the items 1101 to 1108. As an item 1101, duplex print is "available". As an item 1102, N-up print is "available" by the driver. As an item 1103, JOBcopy is available to "10" by the device. As an item 1104, Pagecopy is available to "999". As an item 1105, paper size is "only A4". As an item 1106, OHP insertion is "available" by the driver. As an item 1107, resolution is "both 600/300 dpi are available". As an item 1108, error is "no".

Fig. 12 is a diagram showing an example of a format indicative of adaptivity results which are sent from the server 210 to the client 211 shown in Fig. 2. The diagram shows a format indicative of the adaptivity results which are sent from the server 210 to the client 211 as a result of the retrieval. The data indicates the adaptivity in a certain device and is transferred from the server 210 to the client 211.

In the diagram, reference numeral 1201 denotes an adaptivity and 1202 denotes extended information (EXTEND). Driver correspondence items, error information which is presumed, or the device status is stored in the EXTEND. Reference numeral 1203 denotes a type of EXTEND. For example, "D" indicates a driver adaptivity item, "E" shows an inadaptivity item, and "W" shows an error status.

In Fig. 12, a state where all of eight retrieval items are adaptive. A state where the 4-up print and the OHP insertion are executed in accordance with the ability of the device driver is shown.

Fig. 13 is a diagram showing an example of a retrieval result notification screen which is displayed to the client 211 shown in Fig. 2. A case where it is displayed as a bit map to the client 211 as a retrieval result based on the retrieval condition is shown.

In Fig. 13, reference numerals 1301a to 1301c denote names of the devices; 1302a to 1302c indicate adaptivities and extended information (EXTEND) in the devices. The inadaptivity items, error information, or device status is stored in the EXTEND. By this display, the adaptivities of the devices are sequentially displayed in the order from the high adaptivity and the user can understand which device should be selected.

In the embodiment, it will be understood that the

adaptivity of the LBP 1110 corresponding to the name 1301a of the device is "100" and there are two driver adaptivity items of the printing function of "4 pages" and "OHP insertion". That is, the 4-up print and the
5 OHP insertion print are possible by using the device driver of the LBP 1110.

In an MFP 6550 corresponding to the name 1301b of the device, the adaptivity is "90" and a fact that the inadaptivity item is "duplex" is shown in the EXTEND.
10 In an LBP 3310 corresponding to the name 1301c of the device, it is shown that the adaptivity is "60" and the occurrence of "jam" is shown in the EXTEND.

The embodiment shows a case where the EXTEND is displayed in the order of error, inadaptivity item, and
15 driver adaptivity item.

The embodiment will now be described in detail hereinbelow with reference to flowcharts shown in Figs. 15 to 20.

A registering process for registering into the
20 server by the device will be first explained with reference to the flowchart of Fig. 15.

Fig. 15 is the flowchart showing an example of the first data processing procedure in the device according to the invention. A program based on the flowchart has
25 been stored in the HD 1411 or RAM 1404 in Fig. 14 and is executed by the CPU 1402 in Fig. 14. S1501 to S1504 denote processing steps. The color printer 213 (name:

LBP1110) shown in Fig. 2 will now be described as an example.

It is assumed that according to the LBP 1110, the non-volatile RAM 1412 as self information holding means
5 in the device holds therein position information, a device name, device type name information, and an IP address (hereinafter, attribute information).

When a power source is turned on, the CPU 1402 of the LBP 1110 connects the LBP to the server 210 (step
10 S1501). After the connection, the LBP 1110 reads out the position information and attribute information from the NVRAM 1412 and sends them to the server 210. The server registers those device information into the manager 11 shown in Fig. 1 in the format as shown in
15 Fig. 4.

Subsequently, the CPU retrieves the device driver and if the driver exists in the device, the ability information is obtained from the device driver in the device. The device driver is subsequently retrieved
20 and if it does not exist in the device, the CPU communicates with the device driver in the server 210 or client 211 by using a format 803, thereby obtaining the ability information (refer to Fig. 10) of the device driver. Further, the ability information table
25 shown in Fig. 11 is generated by obtaining the sum of the ability information of the device and the ability information of the device driver (step S1503).

The device itself has obtained the sum of the ability information of the device and the ability information of the device driver here. However, it is also possible to construct the system in a manner such
5 that the device sends the ability information of the device and the ability information of the device driver to the server 210 and the server 210 obtains the sum of the ability information of the device and the ability information of the device driver and registers the sum
10 into the ability information table 15.

After the generated ability information table is sent to the server 210, the CPU 1402 of the LBP 1110 releases the connection to the server 210 (step S1504) and finishes the processes.

15 The ability information table is managed by the manager 11 and referred to by the retrieval unit 12. In the embodiment, the server 210 also adds the status of the device to the information obtained from the device, thereby generating the ability information
20 table 15. By executing such a procedure, each device registers the position information, attribute information, and ability information table into the server 210 after the power-on. Naturally, a procedure similar to that mentioned above is also executed in the
25 case where the updating of the device driver occurs.

Subsequently, processes of the server 210 will be described with reference to the flowchart of Fig. 16.

Fig. 16 is the flowchart showing an example of the first data processing procedure in the server according to the invention. A program based on this flowchart has been stored in the HD 311 or RAM 304 in Fig. 3 and is executed by the CPU 302 in Fig. 3. S1601 to S1609 denote processing steps.

The server 210 is constructed by an event driving type program. When an event is generated, the server analyzes the event and executes processes corresponding thereto.

When the power source is turned on or the event is generated, the server 210 first opens a reception port (step S1601). Subsequently, the server obtains the event (step S1602) and discriminates whether the obtained event is an end command or not (step S1603). As a result of the discrimination, if it is determined that the obtained event is the end command, the reception port is closed (step S1604).

If it is decided that the obtained event is not the end command as a result of the discrimination in step S1603, whether it is a retrieval request or not is discriminated (step S1605). If it is determined that the received event is the retrieval request, a retrieval process is performed in step S1606 and the processing routine is returned to step S1602.

If it is decided that the obtained event is not the retrieval request as a result of the discrimination

in step S1605, whether the obtained event is a registration request for registering the data into the database or not is discriminated in step S1607. If it is determined that the event is the registration request as a result of the discrimination, the received data is registered into a table 400 shown in Fig. 4 (step S1608).

The table 400 in which the received data has been registered is held in the HD in the server 210. In case of registering the ability information, the table is held in the HD 311 and managed by the ability information table 15.

Also in case of registering the device driver, the table is held in the HD 311 and managed by the device driver storage 16.

If it is determined that the event is another request as a result of the discrimination in step S1607, other processes are executed (step S1609) and the processes are finished.

As other processes, there are an inquiry about a location of the device driver and an inquiry about the ability information of the device driver.

Fig. 17 is the flowchart showing an example of the second data processing procedure in the server according to the invention. A program based on the flowchart has been stored in the HD 311 or RAM 304 in Fig. 3 and is executed by the CPU 302 in Fig. 3. S1701

to S1710 denote processing steps. As described in Fig. 16, the server 210 is constructed by an event driving type program. When an event is generated, the server analyzes the event and executes processes corresponding thereto.

When the power source is turned on or when an event is generated, the server 210 first opens the reception port (step S1701). Subsequently, the server obtains the event (step S1702) and discriminates whether the obtained event is the end command or not (step S1703). If it is determined that the obtained event is the end command as a result of the discrimination, the server closes the reception port (step S1704) and finishes the processes.

If it is decided that the event is not the end command as a result of the discrimination in step S1703, whether the event is the retrieval request for retrieving the device driver or not is discriminated (step S1705). If it is determined that the event is the retrieval request, the retrieval process is executed in the device driver registered from the device driver storage 16 in step S1706 and a result of the retrieval process is sent to the device 30. The processes are finished.

If it is determined in step S1705 that the event is not the retrieval request of the device driver, whether the event is the ability information request of

the device driver or not is discriminated in step S1707. If it is decided that the event is the ability information request, the ability information of the device driver (refer to Fig. 10) is obtained from the device driver storage 16 in step S1708. In step S1709, the obtained ability information is sent to the device (for example, device 30) on the requesting source side. The processes are finished.

If it is determined in step S1707 that the event is not the ability information request, processes corresponding to the request are executed as other processes in step S1710. The processes are finished.

The retrieval process in step S1606 which is executed by the server 210 will be described in detail with reference to Fig. 18.

Fig. 18 is a flowchart showing an example of the third data processing procedure in the server according to the invention. A program based on the flowchart has been stored in the HD 311 or RAM 304 in Fig. 3 and is executed by the CPU 302 in Fig. 3. S1801 to S1806 denote processing steps.

In the retrieval process, the retrieval condition set by the client is extracted from a received packet in step S1801. In step S1802, by sending data in the format 802 shown in Fig. 8 to the device 30, the current status of the device is obtained and stored into the ability information table 15. In the

embodiment, the device status and error status correspond.

Subsequently, in step S1803, the ability information table 15 is compared with the retrieval condition, thereby obtaining an adaptivity. For example, adaptivity results of the items shown in Fig. 8 and the LBP 1110 will now be described.

The LBP 1110 has the ability shown in Fig. 9. The device driver of the LBP 1110 has the ability shown in Fig. 10. The sum of both abilities is stored into the ability information table shown in Fig. 11. In step S1802, the current status of the device is also updated.

When comparing with respect to the items of the retrieval condition shown in Fig. 8, DV = printer is "adaptive" with reference to the information 403 shown in Fig. 4, duplex = ON is "available" with reference to the item 1101, N-up is "available by the driver" with reference to the item 1102, JOBCopy is "available" because it is available to "10" by the device with reference to the item 1103, Pagecopy is "available" to "999" for "2" with reference to the item 1104, size is "available" because it is "A4" with reference to the item 1105, OHP = insert is "available by the driver" with reference to the item 1106, and resolution is "available" because "both 300/600 dpi are available" with reference to the item 1107, respectively.

It will be understood that status is "usable status" because error is "NO" by confirming whether the status is "ready" or not with reference to the item 1108.

5 The client 20 is notified of the adaptivity of the device and the device status by using the format shown in Fig. 12 in step S1804 shown in Fig. 18. Since eight items coincide with the eight items requested from the client 20, the adaptivity is set to "100" % and stored
10 into the adaptivity 1201 shown in Fig. 12. The adaptivity item "4-page print" by the driver, and the OHP insertion are inputted into the extended information (EXTEND) 1202. As a type of EXTEND 1203, "D" as a device driver adaptivity item is inputted.
15 When a jam occurs, "jam" is stored into the extended information (EXTEND) 1202.

Whether the adaptivity results of all devices have been obtained or not is discriminated in step S1805. If it is determined that all of the adaptivity results
20 are not obtained yet, the information of the next device is extracted from the table in Fig. 4 in step S1806. The processing routine is returned to step S1802. If the adaptivity results of all of the devices are obtained, that is, the transmission of them to all
25 devices is completed, the processes are finished.

Subsequently, the whole operation in the client 211 will be described.

Fig. 19 is a flowchart showing an example of the first data processing procedure in the client 211 according to the invention. A program based on the flowchart has been stored in the HD 311 or RAM 304 in Fig. 3 and is executed by the CPU 302 in Fig. 3. S1901 to S1907 denote processing steps. The client 211 is constructed by an event driving type program. When an event is generated, the client analyzes the event and executes processes corresponding thereto.

As shown in the flowchart of Fig. 19, the client 211 obtains the event in first step S1901. When the event of the end command is received in step S1902, the processes are finished as they are.

If it is determined in step S1902 that the event is not the end command, for example, the client 211 inputs the retrieval condition from the user with reference to the retrieval condition input screen as shown in Fig. 7. The event is generated when the user clicks a retrieval start button 703.

If it is determined in step S1903 that the event is a request for issuing the retrieval command as mentioned above, the client 211 sends the retrieval request to the server 210 (step S1904).

In this instance, the server 210 is inquired on the basis of the retrieval condition shown in Fig. 8. By storing the retrieval condition onto the HD 311, a reinput of the retrieval condition from the user can be

omitted by using the retrieval condition at the time of the next retrieval.

If it is decided in step S1903 that the event is not the retrieval command issuance, the client 211
5 waits for the retrieval result from the server 210. When the event is received, whether the retrieval result has been received or not is discriminated in step S1905. If it is decided that the retrieval result has been received, for example, when the adaptivity
10 results as shown in Fig. 12 are received, a process for displaying the adaptivity results is executed (step S1906). The processing routine is returned to step S1901.

If it is determined in step S1905 that no
15 retrieval result is received, other processes are executed for other events (step S1907). The processing routine is returned to step S1901.

Although there are the retrieval request of the device driver and the ability information request of
20 the device driver as other events, the processes shown in Fig. 17 corresponding to those requests are executed in a manner similar to those for the server. Naturally, the server 210 is replaced with the client 211.

25 The operation of the device 213 will now be described in detail with reference to a flowchart of Fig. 20.

Fig. 20 is the flowchart showing an example of the second data processing procedure in the device according to the invention. A program based on the flowchart has been stored in the HD 1411 or RAM 1404 in Fig. 14 and is executed by the CPU 1402 in Fig. 14. S2001 to S2005 denote processing steps.

First, the data is received from the server 210 in step S2001. In step S2002, the data is analyzed and whether it is a request command for the current device status (retrieval condition 802 shown in Fig. 8) or not is discriminated. If NO, processes corresponding to the command are executed in step S2003 and step S2005 follows.

If it is decided in step S2002 that the data is a request command for the device status, the information according to the item of the request command is obtained in step S2004. Specifically speaking, the error information and the device status (printing, off-line, etc.) are obtained.

Subsequently, the device status is sent to the server 210 in step S2005 and the processes are finished.

A process for displaying the retrieval results of the client 211 will be finally explained further in detail with reference to a flowchart of Fig. 21.

Fig. 21 is a flowchart showing an example of the second data processing procedure in the client

according to the invention. A program based on the flowchart has been stored in the HD 311 or RAM 304 in Fig. 3 and is executed by the CPU 302 in Fig. 3. S2101 to S2104 denote processing steps.

5 In first step S2101, the adaptivity results of the devices which are sent from the server 210 are received. The adaptivity results are as shown in Fig. 13.

10 In next step S2102, whether the processes of the adaptivity results of all devices have been completed or not is discriminated. If it is determined that they are not yet completed as a result of the discrimination, the processing routine is returned to step S2101 and the adaptivities are received.

15 In step S2102, if it is decided that all processes have been finished, step S2103 follows. The received adaptivity results are sorted in the order from the high adaptivity, thereby deciding the display order. In step S2104, the device names 1301a to 1301c for the
20 devices are displayed with reference to the information 401, the adaptivities, the items serving as extended information 1302a to 1302c, and the other information are displayed for all devices.

25 The device control program on the network according to the invention described above can be executed by a personal computer in accordance with a program which is installed from the outside. In this

case, the invention is also applied to a case where the program is obtained from a memory medium such as CD-ROM, flash memory, floppy disk, or the like or an information group including the program is loaded onto the personal computer from an external memory medium through a network such as Email, personal computer communication, or the like, so that the program is supplied to the personal computer.

Fig. 22 is a diagram for explaining a memory map of a memory medium for storing various control programs according to the invention and shows a case where the memory medium is, for example, a CD-ROM.

In Fig. 22, reference numeral 9999 denotes an area in which directory information has been stored. The directory information indicates a location of a subsequent area 9998 in which an installation program has been stored and a location of an area 9997 in which the network device control program has been stored.

When the network device control program of the invention is installed into the personal computer such as a client 211 or the like, first, the installation program stored in the area 9998 is loaded into the system and executed by the CPU of the personal computer.

Subsequently, the installation program which is executed by the CPU of the personal computer reads out the network device control program from the area 9997

in which the network device control program has been stored and stores it onto the hard disk in the personal computer.

As described above, by combining the ability of
5 the device and the ability of the device driver, the ability of the device can be raised and the advanced function can be provided to the user.

Naturally, after the user selected the devices, the devices which are displayed so that they can be
10 realized by the device drivers need to be set by the device drivers.

(Other embodiments)

Although the embodiment has been described with respect to the case where the selecting reference of
15 the device drivers has been set to the order from the server, the device driver can be also selected from the latest client or the latest device driver. When the updating of the device driver occurs, by executing the present process, the ability information of the latest
20 device driver can be also reflected.

A processing procedure for retrieving a new device driver will now be described with reference to a flowchart of Fig. 23.

Fig. 23 is the flowchart showing an example of a
25 device driver retrieval processing procedure in the server and client according to the invention. S2301 to S2306 denote processing steps.

It is assumed that the present process is started in the case where the device is connected to the network or the case where the device driver is updated.

First, in step S2301, the presence or absence of the device driver in the server and client on the network is identified by using the driver command in the data 803 shown in Fig. 8.

In next step S2302, the server or client having the device driver is identified. In step S2303, versions of the device drivers are sequentially inquired from the relevant terminals.

In step S2304, whether the inquiry processes for all terminals have been completed or not is discriminated. If NO, the next terminal is selected in step S2305.

If the processes for all terminals are finished in step S2304, the ability information of the device driver is inquired from the terminal having the latest device driver and the ability information is obtained in step S2306. The processes are finished.

According to the embodiment, by combining the abilities of the device and the device driver and registering the combined ability information into the server, the adaptivity of the retrieval condition which is sent from the client can be raised. Thus, the result in which the selecting range of the device has been widened and the adaptivity to the retrieval

condition has been raised can be provided to the user.

A construction of a data processing program which can be read out by the network system including the server, client, devices, and the like according to the invention will now be described hereinbelow with
5 reference to a memory map shown in Fig. 24.

Fig. 24 is a diagram for explaining the memory map in a memory medium for storing various data processing programs which can be read out by the network system including the server, client, devices, and the like
10 according to the invention.

Although not shown, there is also a case where information for managing the programs which are stored in the memory medium, for example, version information, implementor, and the like are also stored and
15 information which depends on the OS or the like on the program reading side, for example, icons or the like for identifying and displaying the programs are also stored.

Further, data which depends on the various
20 programs is also managed in the directory. There is also a case where a program for installing various programs into a computer and, if the installation program has been compressed, a decompressing program or
25 the like are also stored.

The functions shown in Figs. 15 to 21 and 23 in the embodiment can be also executed by a host computer

in accordance with a program which is installed from the outside. In this case, the invention is also applied to a case where an information group including the program is supplied to an output apparatus from a memory medium such as CD-ROM, flash memory, FD, or the like or from an external memory medium through the network.

The objects of the invention are also obviously accomplished by a method whereby a memory medium in which program codes of software for realizing the functions of the foregoing embodiments have been recorded as mentioned above is supplied to a system or an apparatus, and a computer (or a CPU or an MPU) of the system or apparatus reads out and executes the program codes stored in the memory medium.

In this case, the program codes themselves read out from the memory medium realize the novel functions of the invention and the memory medium in which the program codes have been stored constructs the invention.

As a memory medium for supplying the program codes, for example, it is possible to use a floppy disk, a hard disk, an optical disk, a magnetooptic disk, a CD-ROM, a CD-R, a magnetic tape, a non-volatile memory card, an ROM, an EEPROM, or the like.

The invention incorporates not only a case where the computer executes the read-out program codes, so

that the functions of the foregoing embodiments are realized but also a case where on the basis of instructions of the program codes, the OS (Operating System) or the like which operates on the computer
5 executes a part or all of the actual processes, and the functions of the foregoing embodiments are realized by those processes.

Further, the invention also incorporates a case where the program codes read out from the memory medium
10 are written into a memory equipped for a function expanding board inserted into a computer or a function expanding unit connected to a computer and, thereafter, a CPU or the like equipped for the function expanding board or function expanding unit executes a part or all
15 of the actual processes on the basis of instructions of the program codes, and the functions of the foregoing embodiments are realized by those processes.

As described above, according to the above embodiments, by combining the ability information of each device connected to the network and the ability
20 information of the device drivers and registering the combined ability information as an ability of the device into the server, the adaptivities of the selectable devices according to the device retrieval
25 condition from the client are presented to the user, so that there are effects such that the selecting range of the devices which can be selected in the whole system

by the user is widened and the desired devices can be freely selected from a plurality of devices which conform with the retrieval condition.

1. The present invention relates to a method for retrieving information from a database, and more particularly to a method for retrieving information from a database in which the user can freely select the desired information from a plurality of information which conform with the retrieval condition.